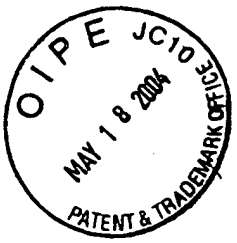
120

17/21



PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant(s)			**Docket No.: 04MV1089**
Hanson et al.			**Examiner: Vo, Lilian**
Serial No.:	**09/422,775**		**Art Unit: 2127**
Filed:	**October 21, 1999**		

Title: METHOD AND APPARATUS FOR AUTOMATIC EXECUTION OF CONCATENATED METHODS ACROSS MULTIPLE HETEROGENEOUS DATA SOURCES

MS Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF (Filed in Triplicate)

An Amendment accompanies this Appeal Brief.

A Petition for a two-month extension of time also accompanies this Appeal Brief.

Please charge the costs for filing this Appeal Brief and the Extension of Time along with any other required costs to Account No. 19-3790 (a duplicate of this page is enclosed).

CERTIFICATE OF MAILING

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to the: Commissioner for Patents, Washington, D.C. 20231,

5/13/04
Date

NATHAN CASS
(Name of person mailing paper)

Nathan Cass
(Signature of person mailing paper)

05/20/2004 HALI11 00000027 193790 09422775
01 FC:1402 330.00 DA

Real party in interest

The real party in interest is Unisys Corporation.

Related appeals and interferences

There are no related appeals or interferences.

Status of claims

All of the presently active claims 1-19, 25-29 and 34-38 in this application are being appealed. Copies of these appealed claims are provided in the attached APPENDIX.

Claims 1-7, 9-11, 13, 14, 16, 17, 25-29 and 34-37 were rejected under 35 U.S.C. 102(e) as being anticipated by Madnick.

Claims 8 and 38 were rejected under 35 U.S.C. 103(a) as being unpatentable over Madnick in view of Brezin and further in view of Yong.

Claim 12 was rejected under 35 U.S.C. 103(a) as being unpatentable over Madnick in view of Dustan.

Claims 15, 18 and 19 were rejected under 35 U.S.C. 103(a) as being unpatentable over Madnick in view of Nierenberg.

Claim 11-19 were rejected under 35 U.S.C. 112, second paragraph.

Claim 38 was objected to under 37CFR 1.75(c).

Status of amendments

An Amendment accompanies this Appeal Brief to overcome the 35 U.S.C.112, second paragraph, rejection of claims 11-19 and also to obviate the objection to claim 38 based on an incorrect dependence recitation. These claim amendments have not been included in the Appealed claims set forth in the Appendix.

Summary of invention

This invention relates to the provision of high speed parallel accessing and operating upon heterogeneous data at a plurality of nodes wherein the heterogeneous data is treated as a single data source object. In response to a request containing a data source object name and also containing a plurality of methods, successive automate execution of these methods can selectively be performed at each respective node and the results transmitted back to a user site.

According to the subject invention, successive methods set out in a script are executed automatically across distributed data, with the results of one execution automatically piped to the next method. An example is the search of a data object followed by an automatic sort of the results of the search, followed by the e-mail of the results of the sort, wherein the data to be searched is distributed across a plurality of nodes, each node having a different type of database.

Figure 1 illustrates a plurality of remote sites or nodes 11, 13, 15, 17 wherein data to be retrieved or accessed is typically spread across the respective nodes. In the illustrative example of Figure 1, the data at node 11 comprises Microsoft NT files, the data at node 13 comprises an Oracle database, the data at node 15 comprises an SQL Server database, and the data at node 17 comprises a Microsoft Access database.

In one example of operation of the system of Figure 1, a user at user site or node 19 propounds a simple request which automatically sets in motion concurrent parallel accessing of all the remote databases 11, 13, 15, 17. The request illustrated in Figure 1 is a search request and the parallel searches are referenced respectively as Search 1, Search 2, Search 3 and Search 4. The searches provide parallel access to the heterogeneous data using a metadata approach and treating the heterogeneous data as if it were a

single object. The simple query or request is first interpreted so as to pass the relevant part of the script from a user node across to the remote nodes. In the embodiment under discussion, queries or requests are presented as JAVA scripts.

In the case of Figure 1, the metadata indicates that the data is contained in the SQL Server, Oracle and/or NT files databases 11, 13, 15 and sets forth the organization of all the data in the respective databases, e.g. the columns and rows and how to interpret the data stored in the database. Accordingly, the user at site 19 does not need to know the data structure and is thus writing applications at a transparent level, i.e., treating the whole network as a single object and writing methods on it.

A special interpreter or "agent" process is employed at the local or user site, which interprets the script/request and "looks up" the appropriate metadata from the NT descriptor file. The local agent then sends appropriate scripts to the particular nodes which contain data corresponding to the data object. An agent (interpreter) module located at each remote node interprets and executes received scripts.

Each agent comprises a module of code (an NT process or the equivalent in another operating system). Thus, two levels of interpretation are employed, a first to interpret the script and a second to interpret and execute the interpreted script at the appropriate nodes. As much processing as possible is performed close to the data, i.e., at the physical sites where the data is stored, in order to minimize message traffic between user and nodes. Thus, a function shipping model is used.

According to the example being discussed in connection with Figure 1, the agent at each remote site, 11, 13, 15, 17 receives the interpreted client request, which includes a data source object name and the methods to be applied, which were originally embedded in the script generated by the user. The remote agent determines from the data source object (1) whether the data is distributed, and if

so, (2) the way in which it is distributed. These details (1) and (2) are contained in the repository 18 of metadata. Once armed with items (1) and (2), the remote agent performs the required method(s) upon the data.

Figure 8 illustrates processing of a script which contains multiple successive or "concatenated" methods. In test 31 of Figure 8, the metadata is checked by the agent at the local site to determine whether the data source is distributed. Test 31 corresponds to test 31 of Figure 2.

In Step 301, the local agent scans the script. In test 303, the local agent determines whether successive methods are included in the script. If not, the routine proceeds to step 35 of Figure 2 of the original application.

If successive methods are involved, the flow proceeds to step 305 where the local agent determines which methods should be performed at the remote sites. This determination is preferably made by accessing a simple table which indicates whether a selected method should be performed remotely adjacent the data or at the user site upon the returned results.

In step 307, the statement is broken into scripts appropriate to the servers at the remote nodes. For example, one may propound the statement:

population.search( ).sort( ).mail( )

to search, for example, the population of the United States for people with particular attributes, sort the results of the search, and then mail the results of the sort. In such case, if the data in "population" were distributed across databases in servers on nodes 1, 3 and 5, the script:

population.search( ).sort( ).

is sent to the servers at each of the nodes 1, 3 and 5. Thus, in this example, the local agent has determined from a table that "search" and "sort" are methods designated for performance at the remote

sites, and has generated an appropriate script to send to each of the sites.

The assistant agent at each of the remote servers on nodes 1, 3 and 5 then interprets the respective script and, on finding the successive methods, search( ).sort( ), performs the first method (search( )) and then leaves the results of that method stored in memory, rather than causing the results to be returned to the coordinating local agent. The second (or further) method(s) are then performed on the results of the earlier method(s), and only when the results of the succession of methods are complete, are the results returned to be merged by the coordinating agent. In this way, if the data object ("population") is distributed, the methods (search, sort) are performed automatically in parallel on the distributed data.

An example of operation of the remote agent is illustrated in Figure 9. The data object "population" 403, 405, 407 is retrieved at each of three respective nodes: Node 1, Node 2 and Node 3. The method "search( )" is performed by the remote agent on each respective data object, producing respective search results 409, 411, 413 stored temporarily in memory at each of the respective Nodes. The remote agent then executes "sort( )" on each of the respective search results, yielding respective sort results 415, 417, 419. The remote agents then transfer the respective sort results to the respective remote messengers, which return them to a coordinating agent at the originating site. The coordinating agent creates the merged results 421 and executes the Mail method to e-mail the final results.

Issues

Whether the Examiner's rejections of claims 1-19, 25-29 and 34-38 are in error.

Grouping of Claims

Group 1 – Claims 1, 25, 29, 34

Group 2 – Claims 2, 26, 31

Group 3 – Claim 6, 7, 27, 28, 36 and 37

Group 4 – Claims 8 and 38

Group 5 – Claim 12

Group 6 – Claims 15, 18 and 19

All of the claims not mentioned above are each in a separate group.

The claims rejected under 35 U.S.C. 102 which are in separate groups are patentably distinct since they include different limitations not met by Madnick in the manner required by 35 U.S.C. 102.

Argument

Rejection of claims 1-7, 9-11, 13, 14, 16, 17, 25-29 and 34-37 under 35 U.S.C. 102(e) as being anticipated by Madnick

It is well established that "A claim is anticipated under 35 U.S.C. 102 only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. V. Union Oil Co. of California, 814 F2nd 628, 631, 2 USPQ2d1051, 1053 (Fed. Cir 1987) (see also MPEP 2131).

Additionally, as held in the recent CAFC decision, Trintec Industries, Inc. v. Top-U.S.A. Corp (CAFC 7/2/02), "Inherent anticipation requires that the missing descriptive material is 'necessarily present,' not merely probably or possibly present, in the prior art." In re Robertson, 169, F.3d 743, 45 49 USPQ2d 1949, 1950-51 (Fed.Cir. 1999).

Still further, the fact that a certain result of characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. In re Rijakaert, 9F.3d, 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993). "In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art. Ex parte Levy. 17 USPQ2d 1461, 1464 (Bd. Pat. Ap. & Inter. 1990) (emphasis in original). (Also see MPEP 2112).

Additionally note the very recent CAFC decision Elan Pharmaceuticals v Mayo Foundation for Medical Education and Research, 68 USPQ2d 1373 (CAFC, Oct. 2, 2003) which holds that: "The disclosure in an anticipating reference must be adequate to enable the desired subject matter. It is insufficient to name or describe the desired subject matter, if it cannot be produced without undue experimentation."

With respect to independent claim 1, Applicant maintains the position that Madnick does not meet this claim in the manner required by 35 U.S.C. 102.

Regarding element (1) of claim 1, Madnick does not disclose, either directly or inherently, the recitation: "propounding a request containing a data source object name wherein the heterogeneous data is treated as a single data source object". In the "Response to Arguments" section of the last Office Action mailed October 10, 2003, the Examiner disagrees and references Madnick, col. 17, lines 35-39 and col. 7, lines 35-44. However, col. 17, lines 35-39 is merely a portion of Madnick's claim 11 which refers to retrieving data from at least one of the heterogeneous data sources and

translating the retrieved data into the data context associated with the request. It is not seen how this can possibly be considered to properly disclose the above cited recitation, either directly or inherently, since this cited recitation requires that the propounded request contain a data source object name wherein the heterogeneous data is treated as a single data source". Note that Madnick, col. 17, lines 35-39 not only does not make any mention of a "single data source object", but also no mention is made of the request. The Examiner's conclusion that "all of the retrieved data is being treated as a single data source after the step of translating" is also not adequate under 35 U.S.C. 102.

Regarding element (3) of claim 1, Madnick does not disclose, either directly or inherently, the recitation: "making a determination as to whether said second method should be performed at the respective nodes or should be performed at the user site after said results are transmitted from each node back to the user site". The Examiner disagrees and refers to col. 2, lines 44-60, col. 15, lines 24-39 and col. 17, lines 1939. It is not seen how these Madnick references can possibly be considered to meet element (3) of claim 1, since these Madnick references do not disclose any counterpart of the first and second methods contained in the claim 1 request, much less disclosing the "making a determination" recitation.

Note that the remaining independent claims 25 29 and 34 are not disclosed by Madnick, either directly or inherently (in the manner required by 35 U.S.C. 102) for the same reasons as provided for claims 1 above. Since all of the independent claims are thus allowable, all of the dependent claims are also allowable.

With regard to the limitations contained in these dependent claims, and the reasons why these specific limitations are

not disclosed or taught by the references, see the arguments presented in Applicant's Amendment filed July 23, 2003.

Regarding the Examiner's withdrawal of the indicated allowability of claims 8 and 38, the rejections thereof fail because of Madnick's lack of teaching of the independent and dependent claims upon which they depend, as explained above. A comparison of these arguments against the "Response to Arguments" in the last Office Action will reveal that the Examiner's rejections do not meet the requirements of 35 U.S.C.102.

Conclusion

In view of the foregoing, Applicant respectfully requests that the Examiner's rejections be reversed.

Respectfully Submitted,
CHARLES ALBIN HANSON

Nathan Cass
Signature of attorney for applicant

NATHAN CASS
Type or print name of attorney

Reg. No. 18,931

Tel: (310) 284-9631
Fax: (310) 284-9634
Address:
Unisys Corp., Law Dept.
2049 Century Park East - Suite 310
Los Angeles, CA 90067

APPENDIX – Appealed Claims

1. A method of accessing and operating upon heterogeneous data at a plurality of nodes comprising the steps of:

(1) propounding a request containing a data source object name wherein the heterogeneous data is treated as a single data source object, said request further containing at least a first method to be performed on the data source object and at least a second method to be performed on the results produced by performance of the first method;

(2) determining whether the data source object is distributed across a plurality of nodes; and

(3) making a determination as to whether said second method should be performed on said results at each respective node or should be performed at the user site after said results are transmitted from each node back to the user site.

2. The method of Claim 1 wherein, if it is determined that the data source object is distributed, and said second method should be performed at the respective nodes, the request is broken into a plurality of new requests, each of said new requests including code representing said first and second methods and having a format appropriate to one of the respective nodes where the data source object resides.

3. The method of Claim 2 further comprising the steps of:

transmitting said new requests to said nodes;

executing the first method concurrently on the data source object at the corresponding nodes;

temporarily storing the results of execution of the first method; and

executing the second method on said results, said step of executing being performed at each of said nodes where the data source object resides.

4. The method of Claim 1 wherein a first agent process at the user site performs the step of making a determination as to whether the second method should be performed at each respective node.

5. The method of Claim 4 wherein, in performing the step of determining whether the data source object is distributed, the first agent process consults a data source descriptor file containing a subset of data contained in a first repository of metadata.

6. The method of Claim 5 wherein a remote agent process automatically executes said first method, automatically stores the results produced by executing said first method, and automatically executes said second method on said results.

7. The method of Claim 6 wherein the results of execution of said second method are automatically returned to the user site and, automatically merged by said first agent process, and wherein a third method is then automatically executed on the merged results by said first agent process.

8. The method of Claim 7 wherein said first, second and third methods respectively comprise a search of the data object, a sort of the results of the search, and an e-mail of the merged results of the search.

9. The method of Claim 5 wherein the data source descriptor file is created from the repository at run-time.

10. The method of Claim 4 wherein a first messenger process cooperates with said first agent process to transmit each said new request to its respective node.

11. The method of Claim 10 wherein said request is in the form of a script and each said new request is in the form of a script having said format.

12. The method of Claim 11 wherein said script and said new scripts are each in the form of a Java script.

13. The method of Claim 11 wherein each of said nodes has associated therewith a respective database, and a respective agent process, each respective agent process comprising code selected to execute the respective new script with respect to the data source object as it is contained in the respective database.

14. The method of Claim 13 wherein each of said databases is different from the remaining respective databases.

15. The method of Claim 14 wherein the respective databases comprise at least two databases, each selected from the following group: Oracle database, NT database and SQL Server.

16. The method of Claim 13 wherein each respective agent process accesses metadata located at the respective node in the course of executing the respective new script at that node.

17. The method of Claim 16 wherein a data source descriptor file is created from the metadata at each respective node for use by the respective agent process.

18. The method of Claim 16 wherein the metadata comprises a collection of data source objects which reflect treatment of data stored in each respective database as a single object and wherein each of said data source objects is broken down into successive class levels.

19. The method of Claim 18 wherein said class levels include a class comprising a System Node, System Server, Data Source Object, Field Desc and System Script.

25. An article of manufacture comprising:

a computer usable medium having computer readable program code means embodied in said medium for accessing and

executing a plurality of methods on data at each of a plurality of nodes, said data being treated as a single data source object, the computer readable code means comprising:

means for receiving a request containing a data source object name wherein heterogeneous data is treated as a single data source object, said request further containing a plurality of methods to be performed on the data source object; and

means for determining whether to execute a second method contained in one of said new requests at a first of said nodes upon the results of execution of a first of said methods at said first node.

26. The article of Claim 25 further including means for determining whether the data source object is distributed across a plurality of nodes; and if the data source object is determined to be distributed, breaking the request into a plurality of new requests, each of said new requests including code representing said second method if said means for determining determines that said second method should be executed at each of said plurality of nodes, each new request having a format appropriate for execution at a respective one of said plurality of nodes.

27. The article of Claim 26 wherein said computer readable code means further contains means for automatically executing said first method, automatically storing the results of the execution, and then automatically executing said second method upon the stored results and returning the results of execution of said second method to a site which transmitted said request.

28. The article of Claim 25 wherein said computer readable code means further includes means for automatically executing a third method included in a said request on the returned results.

29. A data processing apparatus comprising:

means at a user node for receiving a request containing a data source object name wherein heterogeneous data located at a plurality of remote nodes is treated as a single data source object, said request further containing a first method to be performed on the data source object at each of the nodes where said data source object resides and a second method to be performed on the results of performing the first method on the data source object; and

means for determining whether to execute said second method at each of the remote nodes where said data source object resides or at said user node.

34. Computer executable process steps operative to control a computer and stored on a computer readable medium, comprising:

a step to receive a request containing a data source object name wherein heterogeneous data stored at a plurality of nodes is treated as a single data source object, said request further containing a first method to be performed on the data source object and a second method to be performed on the results produced by performance of said first method;

a step to determine whether the data source object is distributed across a plurality of remote nodes; and

a step wherein, if the data source object is determined to be distributed, a determination is made as to whether each of said first and second methods should be performed at the plurality of remote nodes.

35. The process steps of Claim 34 further including:

a step wherein, if it is determined that the data source object is distributed and that said first and second methods should be performed at the remote nodes, said request is broken into a plurality of new requests, each containing code representing said first and second methods.

36. The process steps of Claim 34 further including a step to return results of execution of said second method back to a location where said request originated.

37. The process steps of Claim 36 including a step to merge results received at said location and a step to execute a third method on those results.

38. The process steps of Claim 43 wherein said first, second and third methods respectively are search, sort and e-mail.